# A Performance and Scalability Analysis of the BlueGene/L Architecture

*Kei Davis, Adolfy Hoisie[1], Greg Johnson, Darren J. Kerbyson, Mike Lang, Scott Pakin, Fabrizio Petrini*

Performance and Architecture Laboratory (PAL)
Computer and Computational Sciences Division (CCS)
Los Alamos National Laboratory

## Abstract

Based on a set of measurements done on the 512-node 500MHz prototype and early results on a 2048 node 700MHz BlueGene/L machine at IBM Watson, we present a performance and scalability analysis of the architecture from low-level characteristics to large-scale applications.

In addition, we present predictions using our models for the performance of two representative applications from the ASC[2] workload on the full BlueGene/L configuration of 64K nodes. We have compared the measured values for several of the benchmarks in our suite against the predicted numbers from our performance models. In general, the error bars were relatively low. A comparison between the performance of BlueGene/L and the ASCI Q, the largest supercomputer in the US, is presented, also based on our predictive performance models.

## 1. Introduction

This paper details an initial performance assessment of the BlueGene/L machine being developed by IBM. BlueGene/L is a novel system with high density and low power as key design points and will scale to 64K nodes. It is expected that a 64K-node system with 360 TeraFLOP peak performance will be delivered to Lawrence Livermore National Laboratory. This system will have a footprint of just 64 racks.

The performance analysis that is described here includes:

- System computational noise – The impact of the operating system on the achievable application performance;
- MPI communication characteristics – the performance of near-neighbor, non-near neighbor, hot-spot, and collective communication;
- Single node performance – Memory subsystem bandwidth and latency is analyzed;
- Application performance – Two Los Alamos applications were used to analyze the performance of the system.

Performance models of two Los Alamos applications were used to provide additional analysis. These models provide the capability of accurately predicting the performance of the system without full system hardware being available for measurement.

This paper is structured as follows. Section 2 gives an architectural description of BlueGene/L. Section 3 analyzes the issue of "computational noise" – the effect that the operating system has on the system and application performance. Section 4 describes the performance characteristics of the communication networks. Section 5 deals with single processor performance. Section 6 addresses application performance and scalability, including performance prediction. Most of the results are taken from a 512-node machine running at 500MHz. Also included is a comparison of the predicted performance of BlueGene/L against the performance of ASCI Q and early results from a larger 2048 node BlueGene/L machine clocked at 700MHz. Finally the analysis is summarized in section 7.

## 2. Architectural Description

The basic building block in the BlueGene/L system is a board consisting of 32 nodes. Each node consists of two processors with up to 512 Mbytes of memory and no local disk**.** Each processor is an embedded 32-bit PowerPC 440 and the target clock speed of the final system is 700MHz. Each of the processors can issue two fused multiply-add floating-point instructions per cycle, thus giving a peak performance of 2.8-GF/s. The system-on-a-chip design incorporates the dual processors, three levels of cache, and three interconnects: GigE, tree network, and 3-D torus. The size of the caches are 32KB for the L1 data and instruction cache, and 4MB for the L3 cache. There is also a 2KB prefetch buffer serving the role of a very small L2 cache. There are currently two types of execution modes for each node:

- Communication mode – one processor is dedicated to communication and the other to general processing. This is the default mode.

- Virtual mode – Resources are split between the two processors on the node and each is used as an independent processor.

The packaging in BlueGene/L is very dense, allowing a standard sized cabinet to hold 1024 nodes. The topology of the main communication network is a 3-D torus, which is configured as 32 by 32 by 64 nodes in the full 64K-node system.

The machines that were analyzed in this work were a small 512-node prototype system arranged in an 8 x 8 x 8 3-D mesh

[1] Corresponding author, hoisie@lanl.gov
[2] Advanced Simulation and Computing, formerly ASCI

operating at 500 MHz, and a 2048-node system arranged in a 8x16x16 3-D torus operating at 700 MHz.

Each job on the BlueGene/L system was executed within a *partition* each of which is statically configured based on an XML database. A partition can be as small as eight nodes. Each of these partitions forwards all I/O requests to the I/O nodes on the node card. A node card has up to 32 compute nodes and four I/O nodes. This keeps the kernel on the compute nodes very streamlined. A detailed description of the BlueGene/L architecture can be found elsewhere [1,2,3].

## 3. Computational Noise

*Computational noise* is defined as operating system activity that negatively impacts the processing capability. Computational noise and associated measurement methodology is described elsewhere [4].

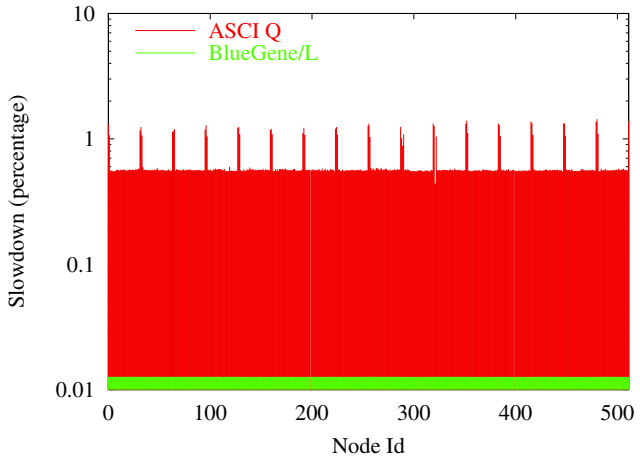


**Figure 1. Slowdown due to computational noise.**

Figure 1 above shows the slowdown due to computational noise on a per-node basis (note the log scale on the Y-axis). A comparison with the computational noise on the ASCI Q machine is also indicated in Figure 1. The level of intrusion of the operating system is minimal in BlueGene/L – it is two orders of magnitude less than both traditional clusters and ASCI Q. The results of this test show that computational noise is negligible on the BlueGene/L, which was expected given the micro-kernel based system software and the tight synchronization of the kernel-level activity.

## 4. Network Performance

In determining the network performance two of the available three networks of BlueGene/L were exercised. The 3-D mesh is used for all communications apart from some of the collectives which utilized the tree network. The basic network performance of the 3-D mesh network is approximately 110 MBytes/s asymptotic bandwidth (Figure 2) and 6µs latency (Figure 3) for nearest-neighbor point-to-point communications in the 500-MHz prototype system. It should be noted that half of the asymptotic bandwidth is achieved at a message of size of approximately 512 bytes. The MPI implementation was stable and didn't have any performance problems.

Figures 4 and 5 show the unidirectional bandwidth and latency seen by processor 0 when communicating to any other node. It is remarkable that the system is so deterministic that it is almost possible to infer the topology of the network. For example, from the bandwidth graph, groups of eight nodes that are aligned on the same row can easily be identified, as well as 8 boards of 64 nodes each. Note that the scale on the *y* axis in Figure 4 ranges only from 109.4 MB/s to 110 MB/s, and that in Figure 5 ranges only from 5.5µs to 8.5µs.

Figures 6 and 7 show the network performance with bi-directional traffic; that is, two neighboring nodes sending messages to each other. The performance is similar to the unidirectional case, and shows that the network and the network interface can handle bi-directional traffic without any performance degradation.

Figure 8 shows the result of the *complement permutation* communication pattern, were each node sends messages to a partner node that is identified by the bit complement of the bit-string representing the sender ID, modulo the total number of nodes. The complement traffic is a good indicator of how the network as a whole behaves under heavy traffic, and whether the actual bisection bandwidth can be achieved under stress. We can see that the pair-wise bandwidth changes according to the location of partners on the topology and the congestion encountered along their communication path. This graph exposes the properties of the 3-D mesh, whose performance is sensitive to process mapping. Note the symmetry as bandwidth from node 0 to node 511 is identical to the bandwidth from node 511 to node 0.

In the *hot-spot* communication, multiple nodes send messages to a single destination. As in the previous traffic pattern, the performance under hotspot is sensitive to the mapping of the hot node in the 3-D mesh. In Figure 9, the hot node is positioned in one of the corners of the 3-D mesh (node 0). The graph shows some interesting properties of the network. With two nodes we get the basic 110 MB/sec, and the performance degrades to 92 MB/sec with 8 nodes because they are aligned on the same row and thus the message routing causes some degree of congestion. With more than 8 nodes the bandwidth increases because it is possible to use more than one incoming link. Nevertheless, the incoming bandwidth never reaches the peak of 330 MB/sec (three times the bandwidth of a single link in each of three dimensions). We speculate that the network interface is not fast enough to receive messages from the three links at full speed.

In the last two graphs of this section we show the performance of the tree network determined by the barrier and broadcast MPI collectives. Figure 10 shows the asymptotic broadcast bandwidth, which is insensitive to the number of nodes. The barrier synchronization latency, shown in Figure 11, is at most 8µs for all the 512 nodes. In both cases the tree network delivers excellent, scalable performance.

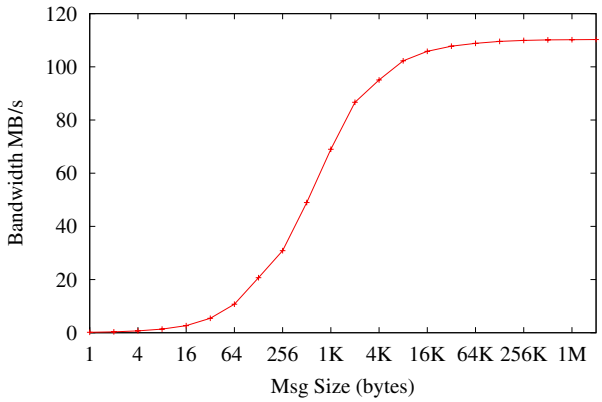


**Figure 2. Unidirectional ping bandwidth.**

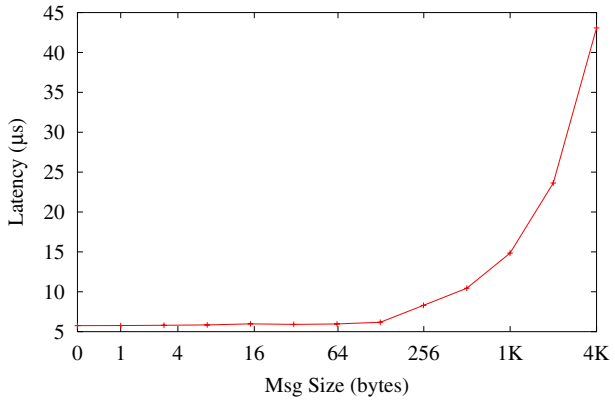


**Figure 3. Unidirectional ping latency.**

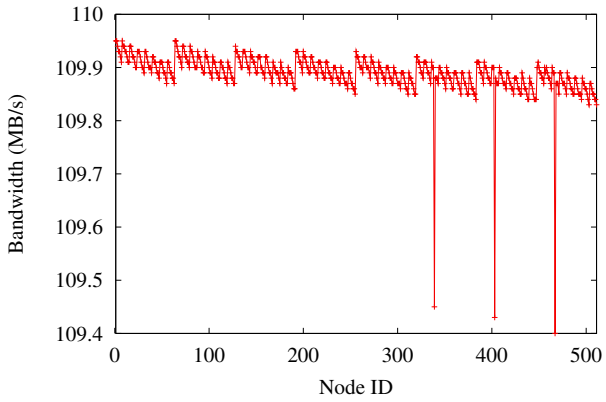


**Figure 4. Unidirectional ping bandwidth, Seen by Node 0.**

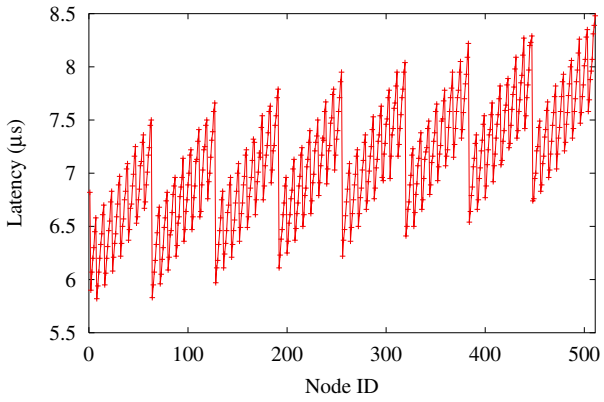


**Figure 5. Unidirectional ping latency, Seen by Node 0.**

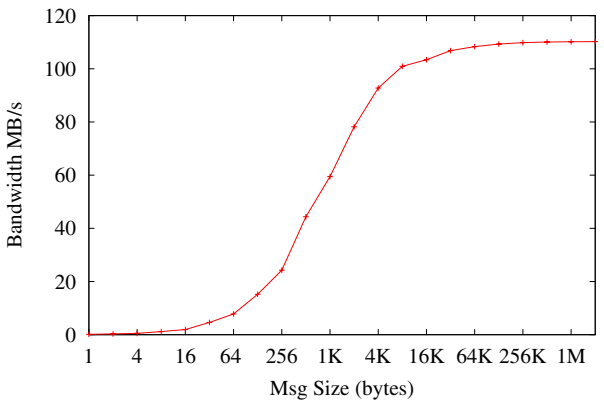


**Figure 6. Bidirectional ping bandwidth.**

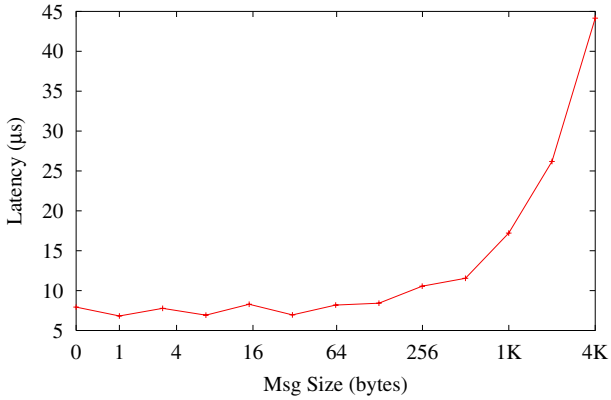


**Figure 7. Bidirectional ping latency.**

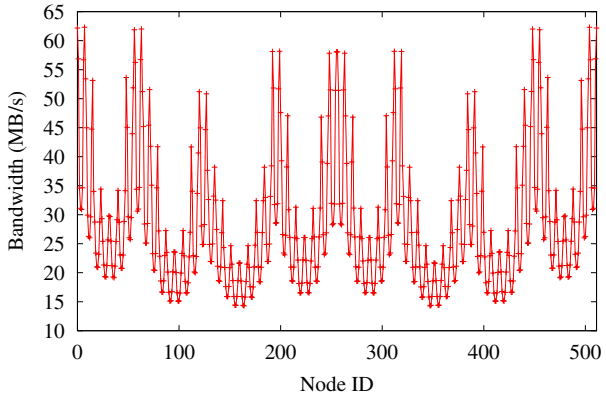


**Figure 8. Complement traffic, Per Node Bandwidth**

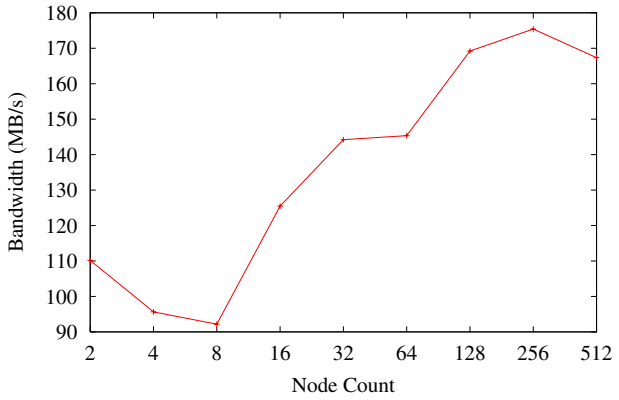


**Figure 9. Hotspot bandwidth.**

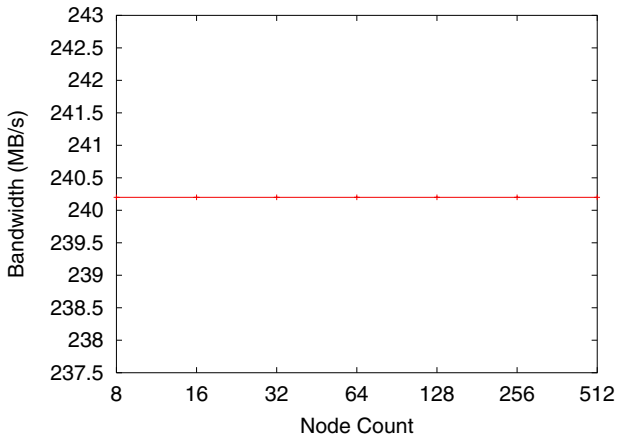


**Figure 10. Broadcast bandwidth.**

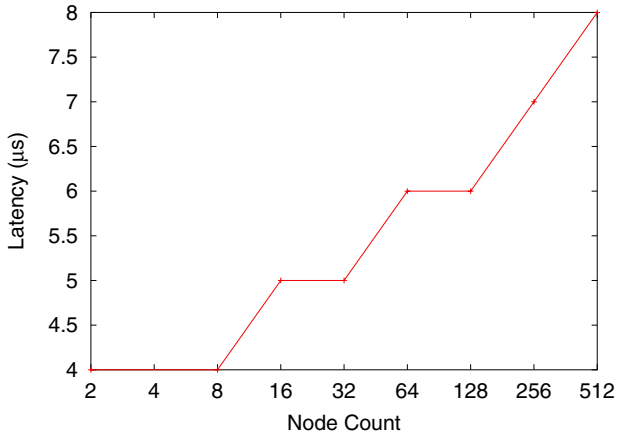


**Figure 11. Barrier performance**

## 5. Single processor benchmarks

Figure 12 shows the results of running CacheBench [7] on a single BlueGene/L processor. Each processor has a peak 5.5 GB/s bandwidth to main memory. For data that fit in cache a 500-MHz BlueGene/L node can read/modify/write up to 3.38 GB/s. For comparison an Opteron processor at 2GHz achieves a peak read/modify/write bandwidth of 10.18 GB/s.

For data that does not fit into the L1 cache, the BlueGene/L processor sees some unusual bimodal behavior in memory bandwidth on the read, write, and read/modify/write tests but not on the hand-optimized (i.e., manually unrolled loops in groups of eight accesses) read, write, and read/modify/write tests. Performance is consistent across runs of CacheBench. The mean read/modify/write bandwidth to main memory is 1.64 GB/s. Again for reference, a 2GHz Opteron achieves 2.98 GB/s.

The latency to memory was measured using *memtime*, a microbenchmark that performs a memory walk with successive memory accesses a cache line stride apart. L1 latency was measured to be 3 cycles, and the latency to L3 34 cycles. Main memory accesses required 85 cycles.

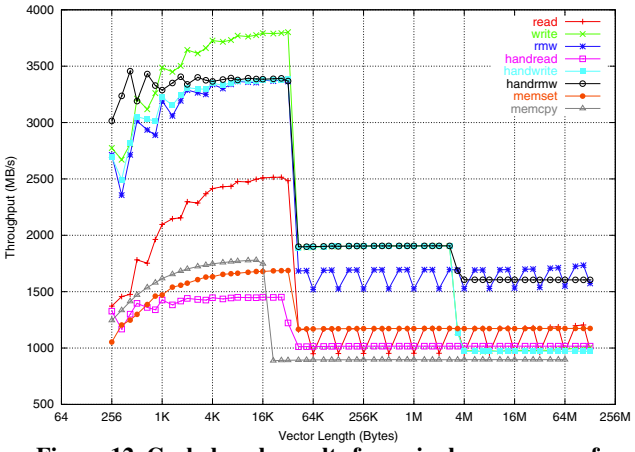


**Figure 12. Cachebench results for a single-processor of BlueGene/L**

## 6. Application Performance

For this paper we utilized two applications that are representative of the ASC workload at Los Alamos. The first application, Sweep3D [5], has characteristics of the computations and communications which consumes the vast majority of the cycles on ASC platforms. The second code, SAGE [6], is an adaptive mesh (AMR) hydro code.

The computational characteristics of these two codes are very different. For instance, Sweep3D is sensitive to the latency of both the memory and of the network, whereas SAGE is equally influenced by bandwidth and latency in the processor and network. All measurements were recorded using the BlueGene/L nodes in communication mode, one processor of the node executing the application while the other processor was reserved to manage communications. Consequently in this data node ID is equivalent to processor ID.

Both of these applications were run in weak scaling mode, that is where the problem size per processor remains the same independent of the number of processors used. Thus the overall problem size grows proportionally with the number of processors; this is typical of the way in which these application are executed.

### 6.1 Sweep3D

Figures 13–16 show the performance and weak-scaling behavior of Sweep3D running on BlueGene/L. We measured performance using two problem sizes per processor (50x50x50 and 5x5x400 cells) and two blocking factors (1 k-plane/1 angle and 10 k-planes/3 angles). Bocking in Sweep3D can significantly change both the surface-to-volume and the processor utilization [5]. Each graph shows two curves: one in which the processor grid is organized with more processors in the *x* dimension than in the *y* dimension and one with more processors in the *y* dimension than in the *x* dimension.

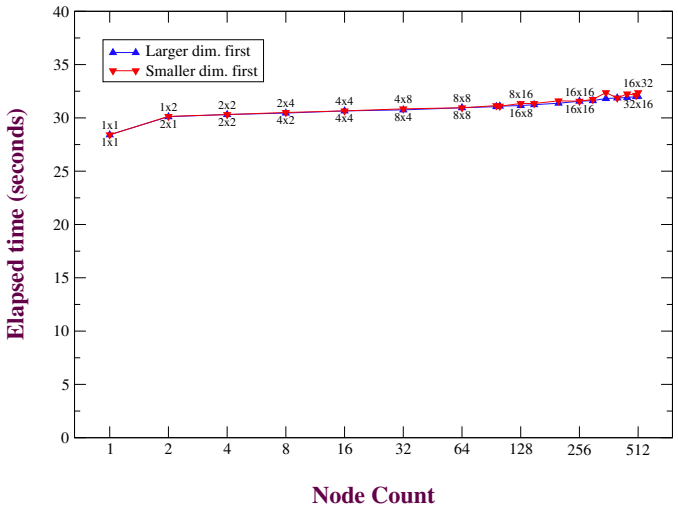


**Figure 13. Sweep 3D, 50×50×50 cells MMI=1, MK=1 blocking,**

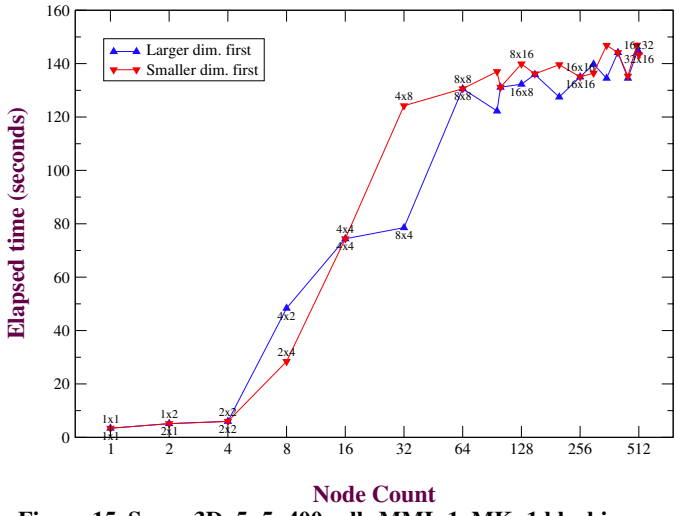


**Figure 15. Sweep3D, 5×5×400 cells MMI=1, MK=1 blocking**

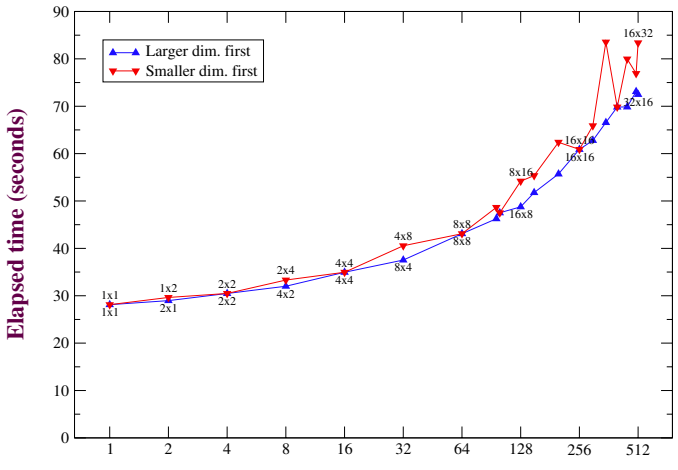


**Figure 14. Sweep3D, 50×50×50 cells,MMI=3, MK=10 blocking.**

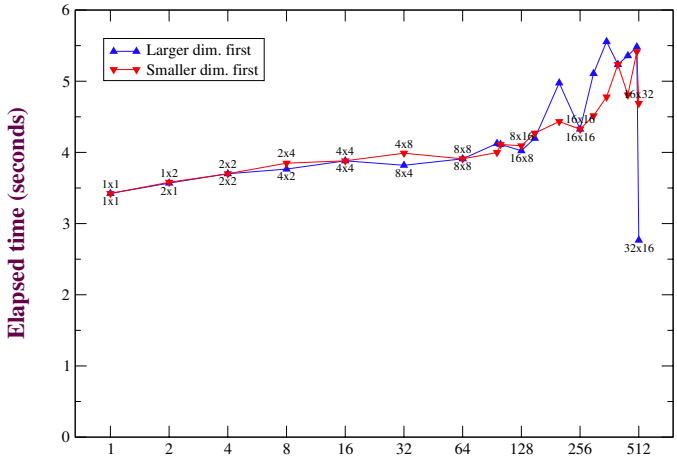


**Figure 16. Sweep3D, 5×5×400 cells, MMI=3, MK=10 blocking.**

The intention of these experiments is to determine how well Sweep3D performs and how well it scales for both coarse- and fine-grained problems with varying message sizes. Also, by changing the processor grid, we can determine the importance of data placement to application performance, an issue of concern given the mesh topology of the machine.

The first experiment, with data presented in Figure 13, represents a coarse-grained problem (125,000 cells and messages of size 400 bytes). In this case the performance scales smoothly from 1 to 512 processors. Furthermore, there is virtually no sensitivity to the shape of the processor grid.

Figure 14 represents the most coarse-grained of all of the Sweep3D runs performed on BlueGene/L. Each processor computes 125,000 cells and sends messages of 4000 bytes apiece. For the most part, this problem size is insensitive to the shape of the processor grid. Although performance degrades at large numbers of processors, this is partly caused by insufficient parallelism within the application and partly by performance characteristics of BlueGene/L.

Figure 15 represents a fine-grained problem. Each processors computes only 10,000 cells but communicates a larger number of messages of 40 bytes apiece. Because of the small computation-communication ratio, these runs are highly sensitive to network performance and communication overhead. The rapid growth in elapsed time is caused by the heavy load on the network. Furthermore, the difference between the two curves conveys sensitivity to the shape of the processor grid. This difference is caused partly by a lack of parallelism within the application and partly by BlueGene/L's handling of heavy message loads.

Finally, Figure 16 represents a fine-to-medium-grained problem that is quite representative of actual usage. Each processor computes 10,000 cells and transmits a number of 400-byte messages. Up to 128 processors, scalability is good and there is negligible sensitivity to the shape of the processor grid. However, elapsed time increases more rapidly from 128–512 processors with the unusual exception of the 32×16 processor grid, most probably caused by a measurement error.

In this section we also present preliminary performance prediction numbers for Sweep3D using PAL's performance model. This has been previously described [5], and has proven to be highly accurate in predicting application performance on all ASC systems to date.

Some of the input parameters for the models are the single processor performance, which for Sweep3D was 3.38% of peak for the 5x5x400 case and 5.1% of peak for the 50x50x50 case, and the value of latency and bandwidth of the communication network as described in Section 4. The blocking parameters, to which the application is very sensitive, are specified in the captions of the respective figures.

Figure 17 shows the measurements vs. the model predictions for the coarse problem 50x50x50, in a weak scaling scenario with a sub-grid of 125K cells per processor. The two blocking schemes have been employed, 10 k-planes and 3 angles per block, or 1 k-plane and 1 angle per block. We see that model predictions are exceptionally good, and the average error is 2.2%. Scaling is very good out to 512 processors for the (10,3) blocking.

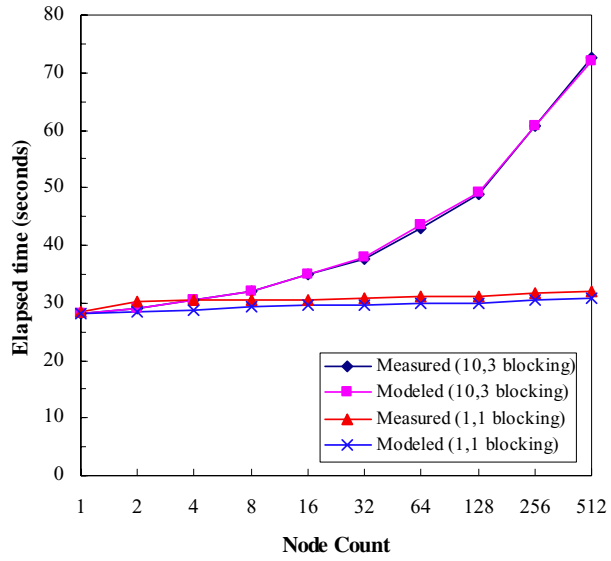


**Figure 17. Measured and Modeled Sweep3D performance for a 50×50×50 subgrid size.**

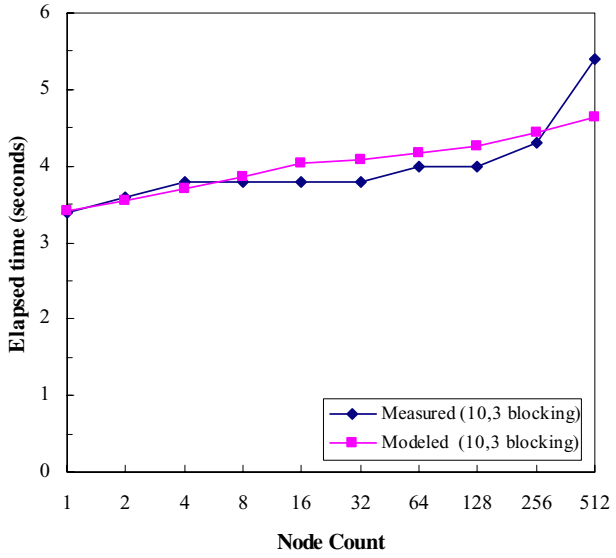


**Figure 18. Measured and Modeled Sweep3D performance for a 5×5×400 subgrid size.**

In Figure 18 a 5x5x400 sub-grid per processor (10K cells per processor in weak scaling) has been utilized for one blocking scheme: 10 k-planes and 3 angles per block. The model predictions are again very good, with an average error of 4.8%. Scaling is also reasonable out to 512 nodes difference at 512 nodes seen the model and measurement is 14%. In Figure 18 (and also Figure 19), 512 nodes was the largest configuration available at the time of testing, and hence further scaling could not be investigated. Possible explanations of the difference between the model and measurement include: lack of torus links which at the largest configuration may result in contention, and lack of adaptive routing for small messages – the case in these runs.

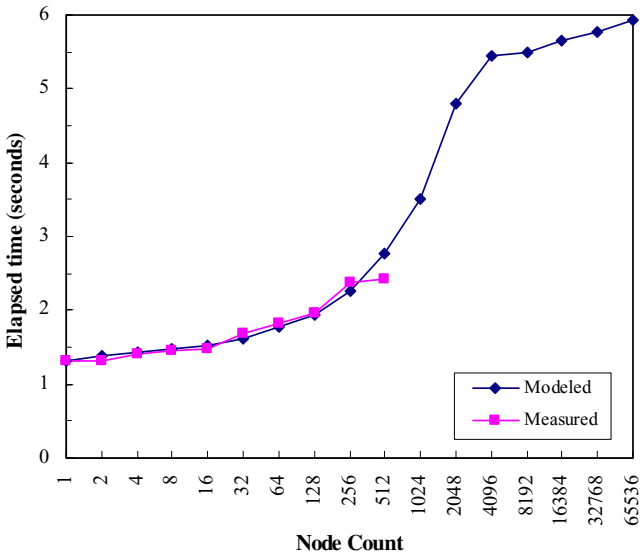


**Figure 19. Measured and Modeled SAGE Performance (timing.input) on BlueGene/L**

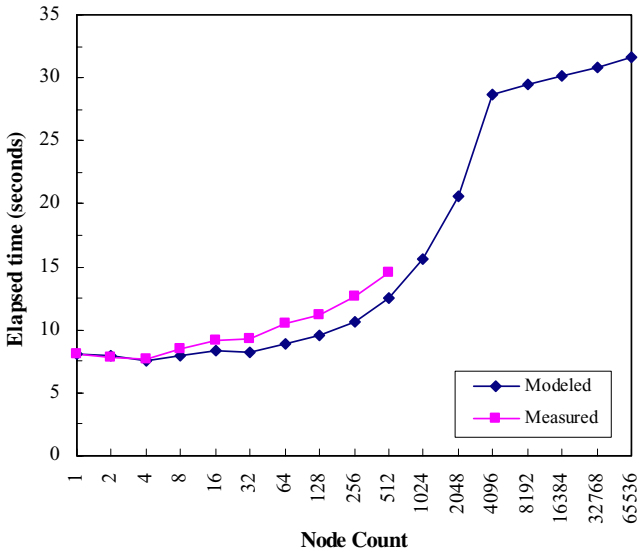


**Figure 20. Measured and Modeled SAGE Performance (timing_h.input) on BlueGene/L**

## 6.2 SAGE

The second application analyzed was SAGE [6]. Figure 19 shows the iteration time of SAGE, both measured and from the SAGE performance model, in a weak-scaling mode for a sub-grid size of 13,500 cells per processor (timing.input). This is a simple case for SAGE that emphasizes scaling and communication aspects, but it is not necessarily realistic because the solver is not included in the computation. We note that measurements match the model very well, and again we are able to use the model to predict out to 64K processors. It can be seen that the efficiency at 64K processors is 25% (1.32s on 1PE vs. 5.14s on 64K PEs).

Figure 20 shows the measured performance of SAGE on BlueGene/L using the timing_h.input deck. This input deck assigns 35,000 cells to each processor in a weak-scaling mode and utilizes the solver in each iteration. The SAGE performance model is also shown in Figure 20 which matches the measurements to within 10%. The model has also been used to predict the performance when scaling the execution of SAGE to 64K processors on BlueGene/L. Note that the performance is expected to level off at 4K processors and above for this input deck to SAGE. This is a characteristic of the application and the position at which this plateau occurs is also a function of the system topology. For instance, this plateau occurs at 512 processors and above on ASCI Q.

## 6.3 Comparisons with ASCI Q

Predictions were made for the performance of a BlueGene/L system containing 64K processors using the Sweep3D performance model which, as shown in Figures 17 and 18, has a high accuracy. Predictions were based on a sub-grid size of 5x5x400 cells per BlueGene/L processor and 12x12x280 cells on an ASCI Q processor (an alpha EV68 running at 1.25GHz). The difference in sub-grid sizes represents the approximate difference in memory capacity per processor between the BlueGene/L and ASCI Q systems. The ASC applications in general are memory limited – they map as large a sub-grid as possible to each processor.

The achievable performance of Sweep3D on BlueGene/L and on ASCI Q per processor is shown in Figure 21. The metric used in this comparison is the number of cells that can be processed per second per processor. The relative performance between BlueGene/L and ASCI Q is shown in Figure 22. The x-axis in these graphs is based on processors rather than nodes since an ASCI Q node contains 4 Alpha processors. A value greater than one indicates a performance advantage of BlueGene/L and a value less than one indicate a performance advantage of ASCI Q. We distinguish two regions in this graph. Up to 8,192 processors the relative performance is shown for an equal processor count on the two systems, whereas above 8,192 processors the number of processors in ASCI Q is fixed at 8,192 (the actual size of the system) while the BlueGene/L processor count increases up to the full 64K configuration.

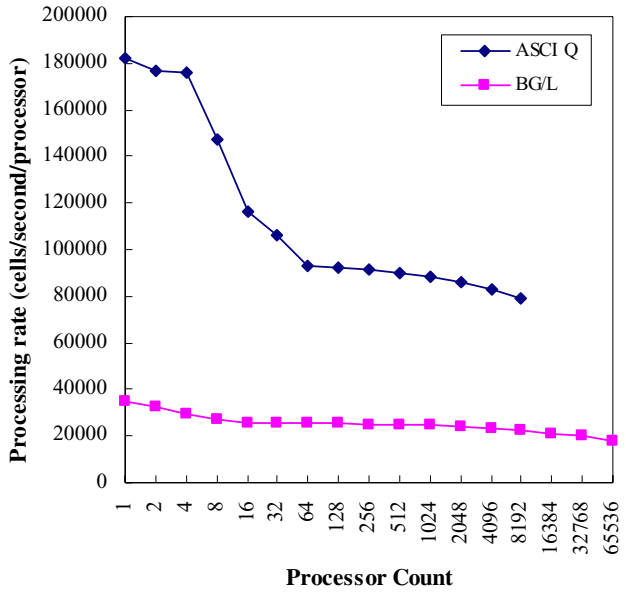


**Figure 21 – Modeled performance of Sweep3D on ASCI Q and BlueGene/L.**

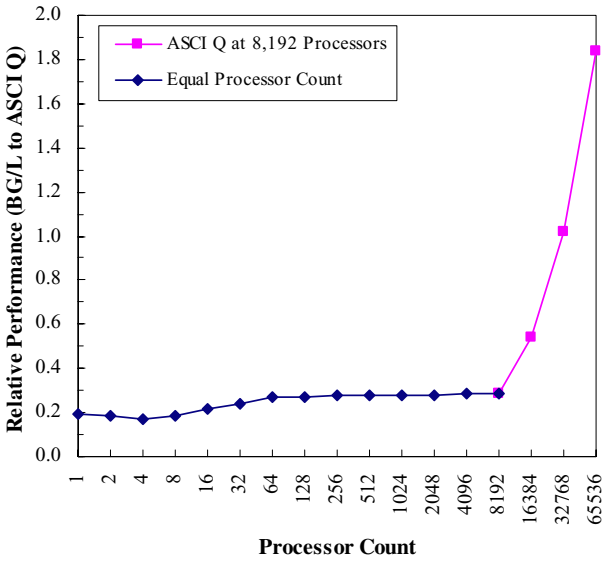


**Figure 22. Relative performance of BlueGene/L to ASCI Q on Sweep 3D.**

For the subgrid sizes under consideration our model indicates that a BlueGene/L machine of size 32K processors would achieve similar performance to ASCI Q using the prototype clocked at 500MHz. A single BlueGene/L processor achieves 0.2 times the performance of an ASCI Q processor. In a similar way, the achievable performance of SAGE on both BlueGene/L and ASCI Q is shown in Figure 23 along with the relative performance of SAGE in Figure 24. In the case of SAGE, a BlueGene/L processor achieves approximately 0.3 times the performance of an ASCI Q processor on this input deck. However, when using over 32K processors BlueGene/L will again outperform the 8,192 processors of ASCI Q on SAGE.

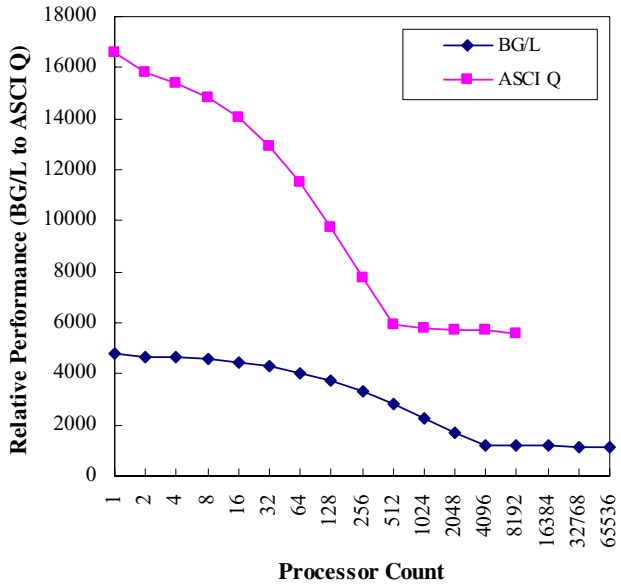


**Figure 23 – Modeled performance of SAGE on ASCI Q and BlueGene/L.**

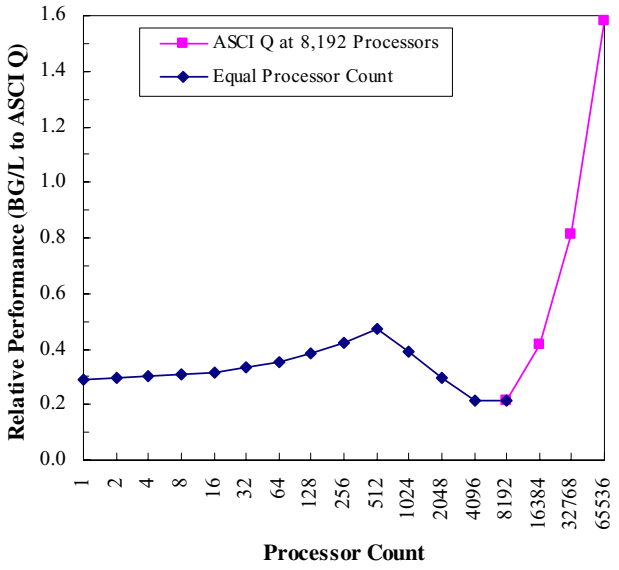


**Figure 24 – Relative Performance of SAGE (timing_h.input) on BlueGene/L to ASCI Q**

## 6.4 Early Results with 2048 nodes of 700Mhz BG/L

Access to the latest BlueGene/L system at IBM Watson was provided to us in June 2004. We report here an analysis of the performance of Sweep3D on a BlueGene/L system comprising 2048 nodes machine at the target operating frequency of 700MHz. This data was measured using the nodes in virtual mode, that is both processors of each node were used to execute application tasks. In Figure 25 we show the relative performance of this BlueGene/L machine in comparison to ASCI Q. The basis of this comparison is the same as that presented in Section 6.3, and the data is directly comparable to that shown in Figure 22.

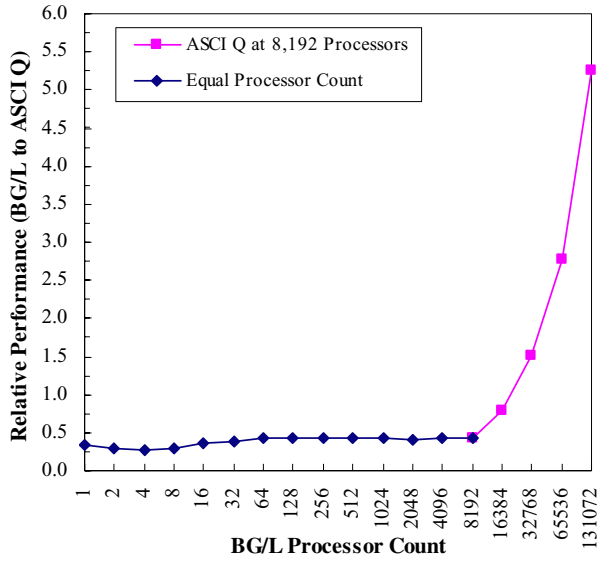


**Figure 25. Relative performance of BlueGene/L to ASCI Q on Sweep 3D (700MHz system).**

It can be seen from the comparison in Figure 25 that the faster BlueGene/L processors achieve a processing rate of approximately 0.3 times that of an ASCI Q processor (up to 8,192 processors). It can also be seen that a BlueGene/L system containing 16,384 processors would achieve the same performance as ASCI Q, and the full-sized BlueGene/L system would be 5.5 times the speed of ASCI Q on Sweep3D.

## 7. Conclusions

We have conducted measurements and analysis, including performance predictions, based on data collected on the prototype 512-node BlueGene/L machine at IBM Watson in December 2003. Additionally we have extended the analysis to include early results from a 2048 BlueGene/L system clocked at the target speed of 700MHz. Special precautions have been exercised to make the data collection as accurate as possible, and the predictions as conservative as is reasonable. By using highly accurate predictive performance models developed by PAL at Los Alamos, we have also compared the performance of the full configuration of 64K processors of BlueGene/L to the performance of ASCI Q machine.

The performance of BlueGene/L is promising. The scalability analysis of Sweep3D and SAGE shown in Section 6 is based on conservative estimates. The system improved with the higher system frequency of 700-MHz (see section 6.4) and the on-going optimization work by the IBM team related to architecture-application mapping

Predictions from 512 processors to tens of thousands of processors include a certain degree of risk, although our models are very accurate. In principle, these uncertainties could affect actual performance both ways. On the one hand a number of features that we exercised with no performance improvement and/or new ones may pan out in the future, and on the other hand scalability could be negatively affected by actual system software and hardware implementations which may take effect at extreme scale.

## Acknowledgments

We wish to thank George Almasi, Jose Castanos, Manish Gupta, Jose Moreira, Martin Ohmacht, Bob Walkup and the rest of the BlueGene/L team at IBM Watson who actively and effectively helped us with this work. This research is supported in part by the Department of Energy's ASC program.

## References

1. An Overview of the BlueGene/L Supercomputer, NR Adiga et al, Proc. of IEEE/ACM SC2002, Baltimore, Maryland, November 16–22, 2002

2. http://www.llnl.gov/asci/platforms/bluegenel/resources.html

3. Unlocking the performance of the BlueGene/L Supercomputer, George Almasi, Sid Chatterjee, Alan Gara, John Gunnels, Manish Gupta, Amy Henning, Jose E. Moreira, Bob Walkup, Proc. of IEEE/ACM SC2004, Pittsburg, PA, November 2004.

4. The Case of the Missing Supercomputer Performance, Achieving Optimal Performance on the 8,192 processors of ASCI Q, Fabrizio Petrini, Darren Kerbyson and Scott Pakin, Proc. of IEEE/ACM SC2003, Phoenix, AZ, November 2003.

5. Performance and Scalability Analysis of Teraflop-Scale Parallel Architectures Using Multidimensional Wavefront Applications. Adolfy Hoisie, Olaf Lubeck, Harvey Wasserman, "The International Journal of High Performance Computing Applications, Sage Science Press, Volume 14, Number 4, Winter 2000.

6. Predictive Performance and Scalability Modeling of a Large-Scale Application, Darren J. Kerbyson, Hank J. Alme, Adolfy Hoisie, Fabrizio Petrini, Harvey J. Wasserman, and Michael Gittings, in Proc. of IEEE/ACM SC2001, Denver, November 2001.

7. Cachebench, available from http://icl.cs.utk.edu/projects/llcbench/cachebench.html